Universidad de La Laguna

E. S. I. T. – INFORMÁTICA
Departamento de Ingeniería Informática y de Sistemas
Programación de Aplicaciones Interactivas

PRACTICA 14: Interpolación mediante Splines

Factor de ponderación [0-10]: 10

14.1. Objetivos

- Ampliar los conocimientos acerca de la programación orientada a objetos (OOP) y el patrón de diseño Modelo-Vista-Controlador (MVC).
- Profundizar los conocimientos de las clases `Graphics` y `Graphics2D`.
- Desarrollar una aplicación interactiva sencilla.
- Incrementar los conocimientos en la programación orientada a eventos.
- Practicar el desarrollo dirigido por pruebas (TDD) y por el comportamiento (BDD).
- Ampliar los conocimientos en interpolación.

14.2. Descripción

En el análisis numérico, un *Spline* [1] es una función especial definida a trozos, donde cada trozo viene especificado por un polinomio de cierto grado. Dados $n+1$ puntos o nodos $(x, y = f(x))$, se desea llevar a cabo una interpolación de la función $f(x)$ entre todos ellos mediante una curva lo más suave posible. Para ello, se podría utilizar un polinomio de grado n, pero generalmente la curva que se obtendría sería demasiado ondulada y poco suave. La idea de los Splines es que se puede calcular un polinomio para cada uno de los n sub-intervalos establecidos por los $n+1$ puntos, de manera que en conjunto parezcan que forman parte de un único polinomio interpolador.

De manera formal, dados $n+1$ puntos o nodos $\{(x_i, y_i = f(x_i)) : i = 0, 1, \ldots, n\}$ se debe llevar a cabo una interpolación entre cada par de puntos (x_i, y_i) y (x_{i+1}, y_{i+1}) mediante polinomios $y = q_i(x), i = 1, 2, \ldots, n$. Para ello, generalmente, se suelen utilizar polinomios de grado 3, dando lugar a los conocidos *Splines cúbicos* (véase la Figura 14.1). El problema, por tanto, reside en **determinar los n polinomios $q_i(x) : i = 1, \ldots, n$ de grado 3 dados $n+1$ puntos de entrada**.

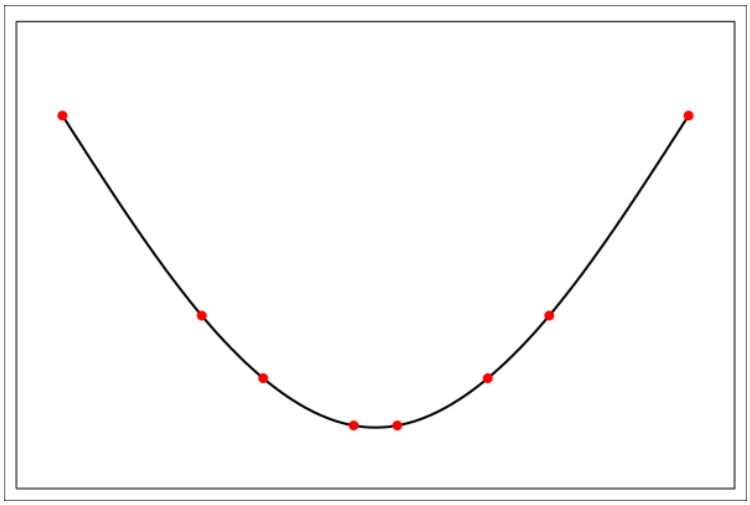

Figura 14.1: Interpolación a través de $n + 1 = 8$ puntos mediante un spline cúbico. Cada uno de los $n = 7$ polinomios que interpolan entre cada par de puntos es de grado 3.

En esta práctica, deberá desarrollar un programa `Splines.java` que implemente el algoritmo de interpolación mediante Splines cúbicos descrito en [2], en concreto, las ecuaciones 1–4. Nótese que, para calcular cada polinomio $q_i(x) : i = 1, \ldots, n$, además de conocer los valores x_{i-1}, x_i, $y_{i-1} = q_i(x_{i-1})$ e $y_i = q_i(x_i)$, también deberán conocerse los valores $k_{i-1} = q_i'(x_{i-1})$ y $k_i = q_i'(x_i)$, es decir, el valor de la primera derivada de la función $q_i(x)$ en x_{i-1} y x_i, respectivamente. En el caso de esta práctica, dichos valores k_{i-1} y k_i podrán ser fijados arbitrariamente, aunque cabe mencionar que dependiendo de los valores indicados, la curva será más o menos suave.

Existen diferentes aplicaciones que ilustran el funcionamiento del anterior algoritmo, como por ejemplo, las accesibles a través de [3-4]. Las siguientes deberán tomarse como especificaciones del programa a desarrollar:

- Deberá utilizar el patrón de diseño MVC y poner especial énfasis en las buenas prácticas y principios de OOP.
- Deberá aplicar el paradima de desarrollo dirigido por pruebas (TDD) y por comportamiento (BDD), esto es, deberá establecer un conjunto de pruebas que permitan comprobar el comportamiento esperado de la aplicación bajo ciertas circunstancias, para a continuación, implementar dicho comportamiento con el objetivo de que el resultado del conjunto de pruebas sea satisfactorio.
- El programa a desarrollar ha de funcionar como aplicación Java autónoma y también como *applet*.
- En su interfaz, existirán dos paneles. El primero de ellos permitirá interactuar con el programa, a través de campos de texto y botones. En el segundo se mostrará el Spline y los $n + 1$ nodos que forman parte del mismo. El Spline deberá mostrarse como una curva continua de un color determinado, mientras que los nodos vendrán

representados por puntos de un color diferente al de la curva y de un grosor superior al de la misma.

- Al iniciar la aplicación, el contenido del segundo panel estará vacío. En el primer panel, existirá un campo de texto que permitirá indicar el número de nodos ($n+1$) del Spline. Además, existirá un botón *Generar* que permitirá obtener de manera aleatoria dichos $n+1$ nodos. Durante la generación aleatoria, se deberá cumplir que $x_0 < x_1 < \ldots < x_n$. De manera inmediata, una vez generados los $n+1$ nodos, el programa deberá calcular los n polinomios $q_i(x) : i = 1, \ldots, n$ y dibujar el Spline que corresponda en el segundo panel.

- Una vez generado el Spline inicial, un nodo determinado podrá ser seleccionado mediante el teclado para cambiar su posición dentro del panel. Un cambio de posición de un nodo conllevará calcular los n polinomios del Spline y repintar dicho Spline en el panel inmediatamente. Además, se deberá seguir manteniendo la restricción $x_0 < x_1 < \ldots < x_n$, es decir, un nodo determinado no puede cruzarse horizontalmente con sus nodos adyacentes.

- En el primer panel, también existirá un botón *Reiniciar* que limpiará el panel donde se dibuja un Spline determinado.

- Por último, el primer panel dispondrá de un icono *Información* que al ser pulsado abrirá una nueva ventana conteniendo información sobre la aplicación y su autor.

- La interfaz de usuario no incorporará ningún otro elemento ni funcionalidad que las descritas en esta especificación.

14.3. Referencias

[1] Wikipedia: ”Spline (mathematics)”.
`https://en.wikipedia.org/wiki/Spline_(mathematics)`

[2] Wikipedia: ”Spline Interpolation”.
`https://en.wikipedia.org/wiki/Spline_interpolation`

[3] MooTool/HTML5 Cubic Spline demo - Numerical Recipes in C 2nd Edition, Press, Teukolsky, Vetterliong, Flannery, pg. 113.
`http://ctyeung.com/js/MooTool/CubicSpline/CubicSpline.html`

[4] Protovis - Spline Editor
`http://mbostock.github.io/protovis/ex/splines.html`